

NLang, an alternative to RDF

[Victor Porton](#)

Victor Porton © 2014. See [the License](#)

[Jul 21, 2014 21:34:26](#)

1. Rationale

Consider the following example fragments of RDF (in Turtle format).

First file:

```
:Alex :sex :male .  
:Alex :age 10 .
```

Second file:

```
:Alex :sex :female .  
:Alex :age 15 .
```

If we load these two files, it could create a total mess of two Alexes.

In my new proposed language NLang it would be instead the following n -ary (in the example binary) predicates (for these n -ary predicates the letter “N” in NLang):

```
:Alex [  
    :sex :male ;  
    :age 10  
].
```

and

```
:Alex [  
    :sex :female ;  
    :age 15  
].
```

Now it is clear that these are two different Alexes (even despite they share the same object URL :Alex).

This issue becomes important in real world applications which load more resources files from the Web during program execution.

As a realization of the above idea, we define objects. With each namespace they are associated zero or more *objects*.

Example of NLang (similar but different to RDF/Turtle grammar):

```
# Suggestion to load http://example.org/module  
# if we need more information on the object http://example.org/example2  
# which is expected to be of the class :transformation  
<http://example.org/module> [  
    a nlang:seeAlso ;  
    nlang:object <http://example.org/example2> ;  
    nlang:class :transformation
```

```

] .
<http://example.org/example> [
  a :transformer ;
  dc:description <http://...> ;
  # Other Dublin Core metadata.
  :source-namespace <...> ;
  :target-namespace <...> ;
  :precedence <...> ;
  # Creates zero or more :scripts attributes, with their values being objects
with <http://example.org/scripts> URL:
  :scripts ref <http://example.org/scripts>
] .

```

```

<http://example.org/scripts> [
  a :xslt ;
  :version "2.0" ;
  :script-url <http://example.org/scripts/foo.xslt> ;
  :transformer-kind :entire ;
  :argument [
    :name "debug" ;
    :value false
  ] ;
  :argument [
    :name "other" ;
    :value 123
  ] ;
  #:initial-context-node ... ; # See XSLT 2.0 spec.
  :initial-template "first" ;
  :initial-mode: "first" ;
  :completeness 0.9 ;
  :stability 0.9 ;
  :preference 0.9
] .

```

2. Formal grammar

TODO

3. Semantics

TODO

4. Representing RDF in NLang and NLang in RDF

TODO

<http://martinfowler.com/bliki/CQRS.html> and <https://issues.apache.org/jira/browse/MARMOTTA-435>

<https://code.google.com/p/jrdf/wiki/GlobalGraphs>

5. A proposed implementation

I am going to implement it in Ada, probably using AdaGOOP: <http://adagoop.martincarlisle.com/>